Engineering Data Analysis with Matlab

Anke Scherb

Engineering Risk Analysis Group
Technische Universität München

Phone: 089 289 23013
E-mail: anke.scherb@tum.de

Engineering Data Analysis with Matlab

For your project work …

- Please form groups of 2-3 students for your project work
- Send an email to anke.scherb@tum.de or tell the tutors during the tutorial **until May 9**
- This is requested for receiving the ECTS !

Engineering Data Analysis with Matlab

Today's lecture

- Matlab operators
- Basic programming
- Plotting data

Matlab – Operators

Relational operators

- Compare operands quantitatively
- Element-by-element comparisons between two arrays or matrices
- Return logial array of the same size as operands, with elements set to logical 1 (true) where the relation is true, and elements set to logical 0 (false) where it is not

Matlab – Operators

Relational operators

<	Less than
<=	Less than or equal
==	Equal
>=	Greater than or equal
>	Greater than
~=	Not equal

Matlab – Operators

Relational operators - example

```
>> A = [4 5; 6 7 ]
A =
     4     5
     6     7
>> B = B = [7, 9; 2, 3]
B =
     7     9
     2     3
>> C = A > B
C =
     0     0
     1     1
```

Example: RelOperators.m

Matlab – Operators

Find elements that meet a condition

```
>> A=randi(25,5)
A =
     7    14     7     5    21
    13     4    21     7    15
    18     4     7    16    14
    23     7    24    12    23
    24    22     9     9     8

>> B= A > 12
B =
     0     1     0     0     1
     1     0     1     0     1
     1     0     0     1     1
     1     0     1     0     1
     1     1     0     0     0
```

Matlab – Operators

Find elements that meet a condition

```
>> c=A(B)
c =                    % c contains all elements of A that
    13                     are larger than 12
    18
    23
    24
    14
    22
    21
    24
    16
    21
    15
    14
    23
```

Matlab – Operators

Find elements that meet a condition

```
>> D=find (A > 12)
D =      % D contains all positions of elements of A
     2                    larger than 12
     3
     4
     5
     6
    10
    12
    14
    15
    16
    17
    18
    19
```

Matlab – Operators

Select elements with special criteria

```
>> v = [4, 8, 19, 22, -4 10]
v =
     4     8    19    22    -4    10
>> w = [4, -7, 19, 22, -8, 20]
w =
     4    -7    19    22    -8    20
>> x = v > w
x =
     0     1     0     0     1     0
>> y = v(x)    % select values from vector x
y =
     8    -4
```

Matlab – Operators

Logical operators

`a & b`	And
`and(a,b)`	And
`a | b`	Or
`or(a,b)`	Or
`~a`	Not
`not(a)`	Not

Note: Logical operations can be applied element-wise

`any(A)`	Returns true if at least one element is true and false else

Matlab – Programming tools

For loop – Execute statements a specified number of times

```
for index = values
   statements
end
```

```
for i = 1:k
   for j = 1:k
      A(i,j) = i+j;
   end
end
```

Matlab – Programming tools

While loop – Repeatedly execute statements while condition is true

```
while expression
   statements
end
```

```
i = 1;
while i < 10
   v(i) = i^2;
   i = i+1;
end
```

Example: loops.m

Matlab – Programming tools

If, elseif, else statement – Execute statements if condition is true

```
if expression
   statements
end

if expression
   statements
elseif expression
   statements
else
   statements
end
```

Example: logif.m

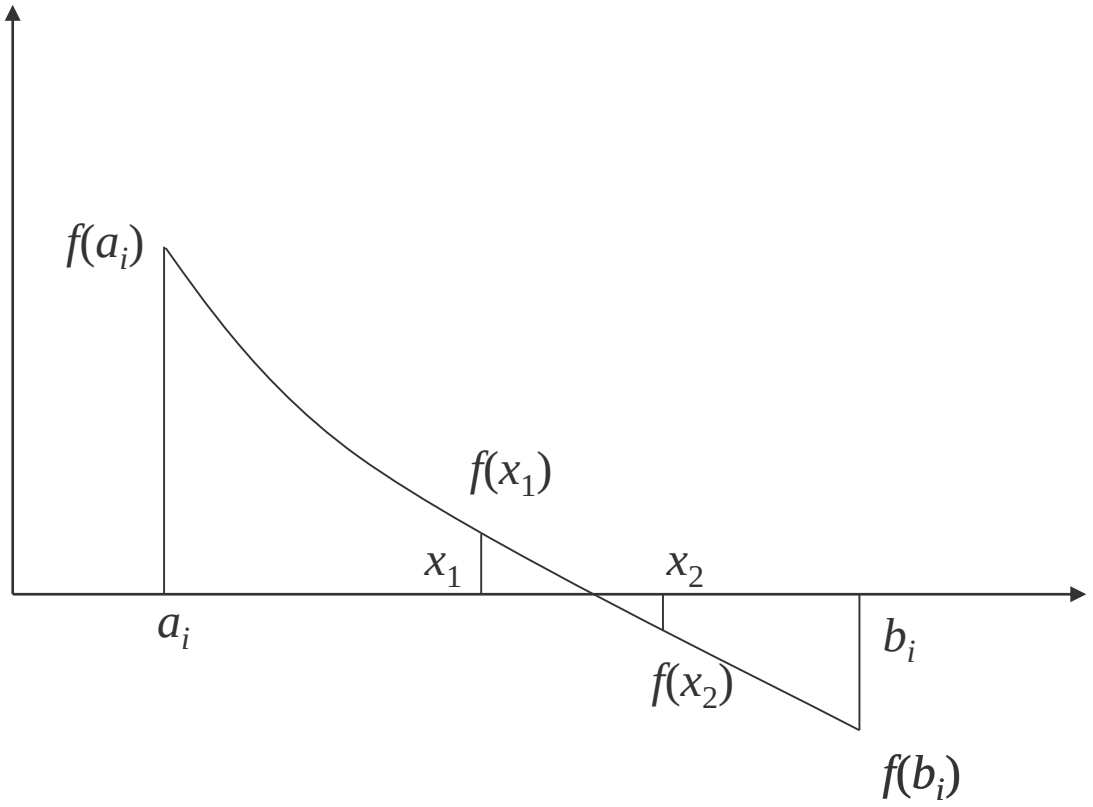
Matlab – Programming tools

Break statement – Exit a loop

```
n = 0;
for i = 1:k
   n = n+i;
   if n >= 10
      break;
   end
end
```

Matlab – Programming tools

Example – Bisection method for finding the root of a function



$x_i = (a_i + b_i)/2$

If $f(x_i)\, f(a_i) < 0$

$b_{i+1} = x_i$

Else

$a_{i+1} = x_i$

Example: bisection_method.m

Matlab – Plotting

Example data plot types:

- Data plot
- Surface plot
- Scatter plot
- Cone plot
- Bar graph
- Errorbars
- Pie chart
- …

Matlab – Plotting

Plotting commands

`figure`	Create figure object
`hold`	Retain current graph when adding new graphs
`plot`	2D line plot
`xlim, ylim`	Set axes limits
`legend`	Create graph legend
`xlabel, ylabel`	Set axes labels

Matlab – Plotting

Simple plot – Example

Plot (x, y)

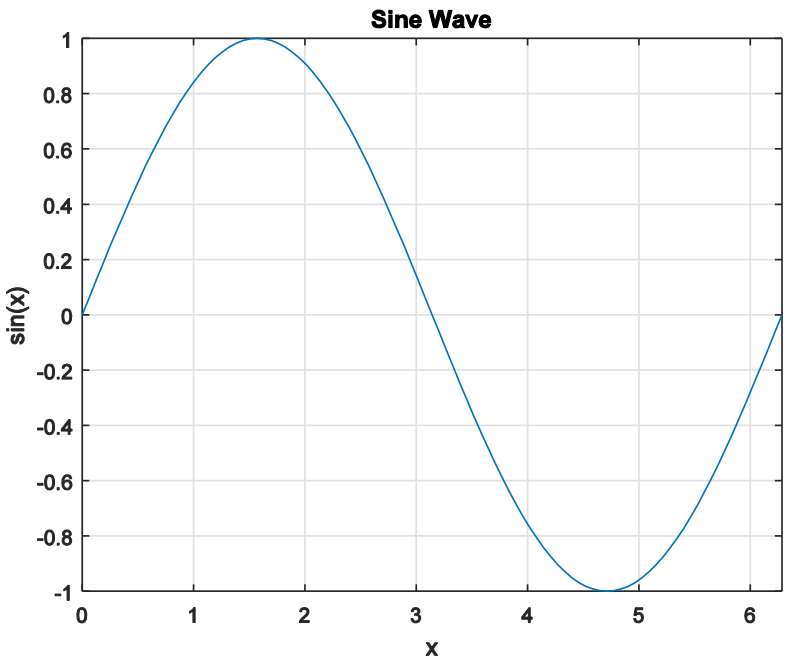
```
>> x=[0:pi/50:2*pi];      % Set domain values
>> y=sin(x);              % Set range values
>> plot (x,y);            % Plot data
>> grid;                  % Turns grid on
>> title 'Sine Wave';     % Sets title
>> xlabel 'x';            % Sets label of x axis
>> ylabel 'sin(x)';       % Sets label of y axis
>> xlim([x(1) x(end)]); % Sets x plot limits
```

Matlab – Plotting

Simple plot – Example

Plot (x, y)

```
>> x=[0:pi/50:2*pi];      % Set domain values
>> y=sin(x);              % Set rang
>> plot (x,y);            % Plot dat
>> grid;                  % Turns gr
>> title 'Sine Wave';     % Sets tit
>> xlabel 'x';            % Sets lab
>> ylabel 'sin(x)';       % Sets lab
>> xlim([x(1) x(end)]); % Sets x p
```



Matlab – Edit Graphics Style

```
>> plot(x, y, '<color> <point style> <line style>')
```

Color		Point style		Line style	
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star		
y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Matlab – Plotting

Multiple plot – Example

Plot (x, y)

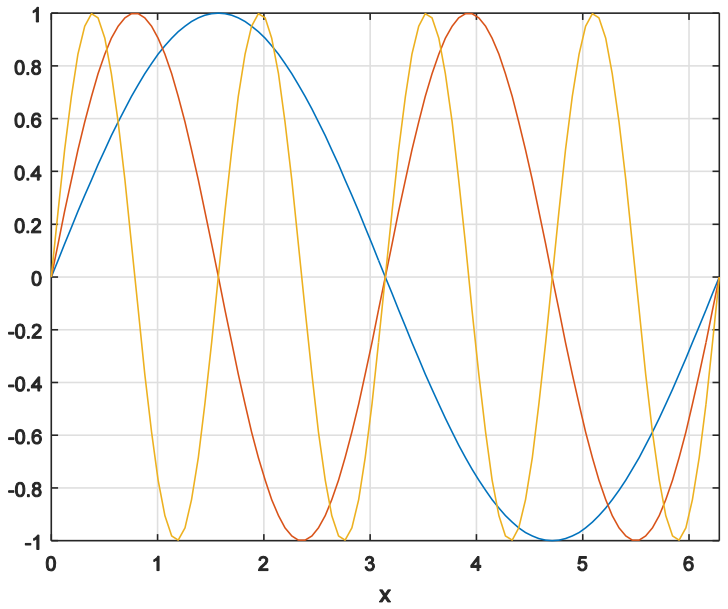
```
>> x=[0:pi/50:2*pi];                    % Set domain values
>> y1=sin(x);                           % Set range values
>> y2=sin(2*x);
>> y3=sin(4*x);
>> plot (x, [y1; y2; y3]);              % Plot data
>> grid;                                % Turns grid on
>> xlabel 'x';                          % Sets label of x axis
>> xlim([x(1) x(end)]);                 % Sets x plot limits
```

Matlab – Plotting

Multiple plot – Example

Plot (x, y)

```
>> x=[0:pi/50:2*pi];       % Set domain values
>> y1=sin(x);              % Set range values
>> y2=sin(2*x);
>> y3=sin(4*x);
>> plot (x, [y1; y2; y3]);       %
>> grid;                         %
>> xlabel 'x';           % Sets labe
>> xlim([x(1) x(end)]); % Sets x pl
```



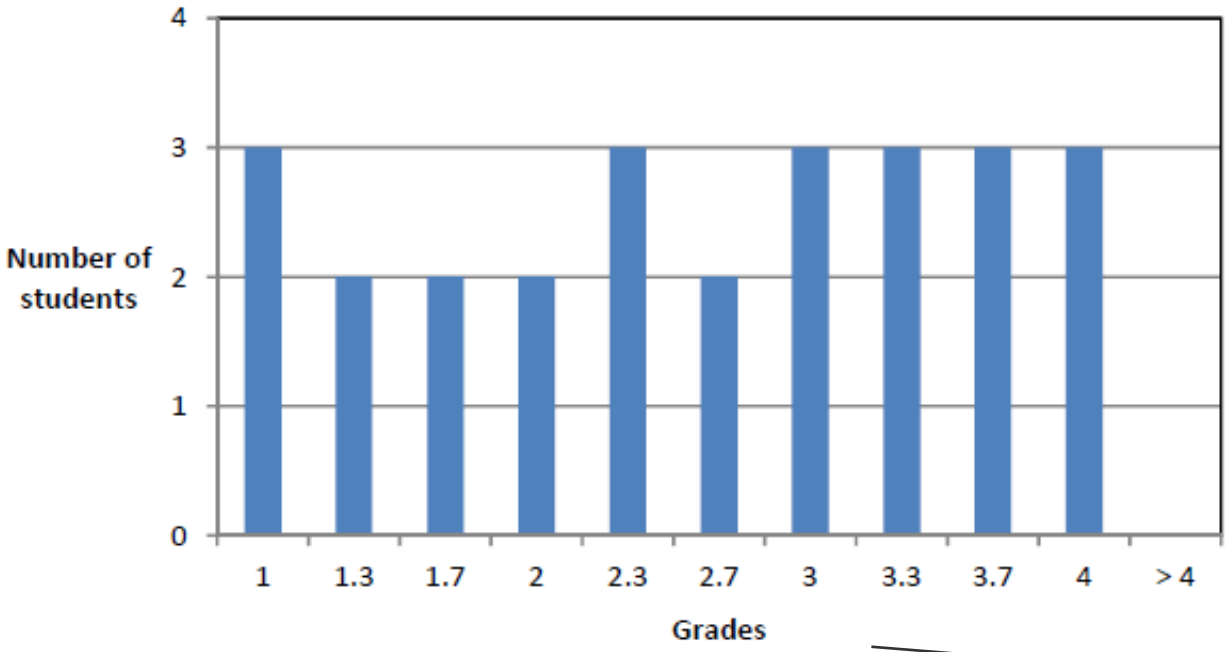
Example: plotting.m

Graphical representation of data sets

Bar chart – Representation of discrete quantities or categorical data

```
bar(data)
```

Bar chart of grades in an exam



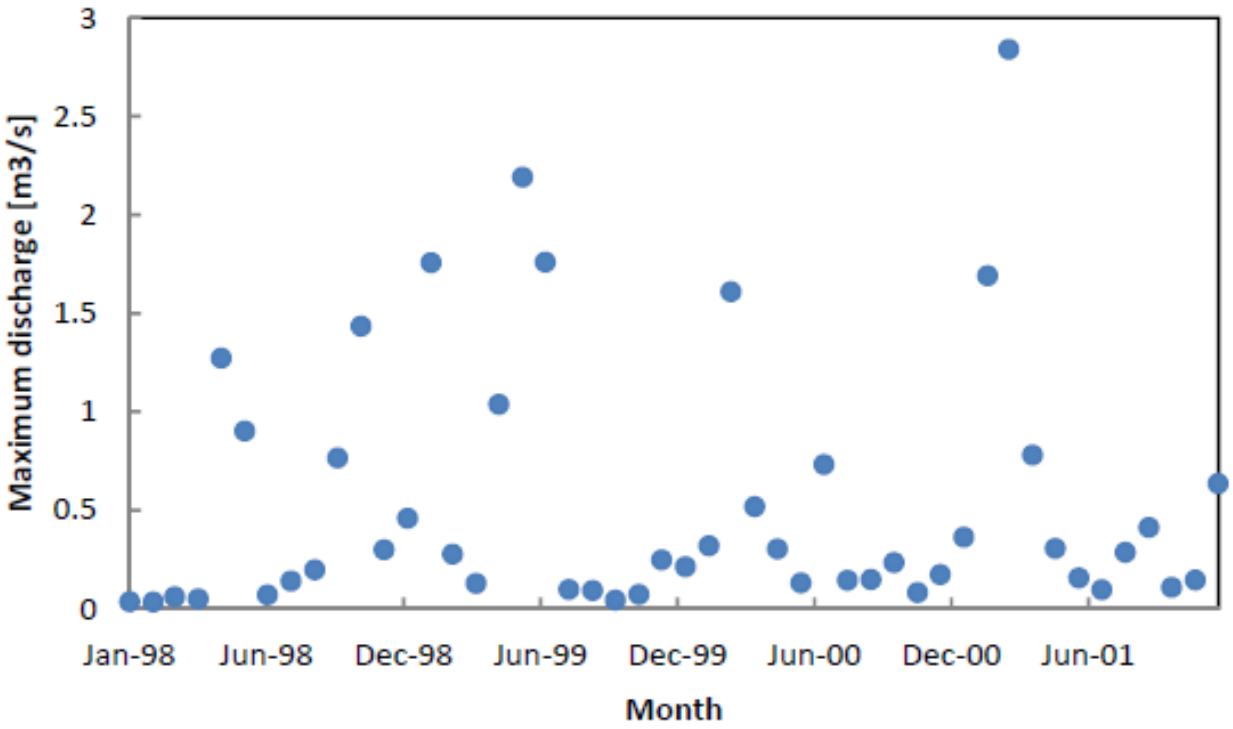
Number of occurences

Discrete states

Graphical representation of data sets

Series diagram – Representation of continuous data

Series diagram of maximum monthly discharge in a river

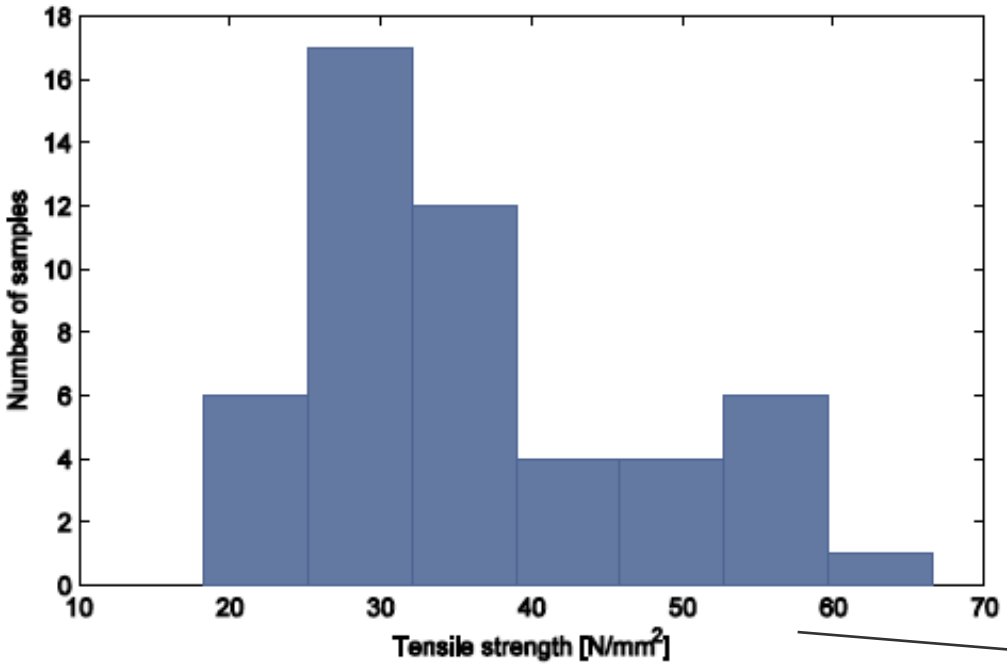


Graphical representation of data sets

Histogram – Representation of large continuous data sets

```
hist(data,nbins)
```

Histogram of tensile strength data



Number of occurences

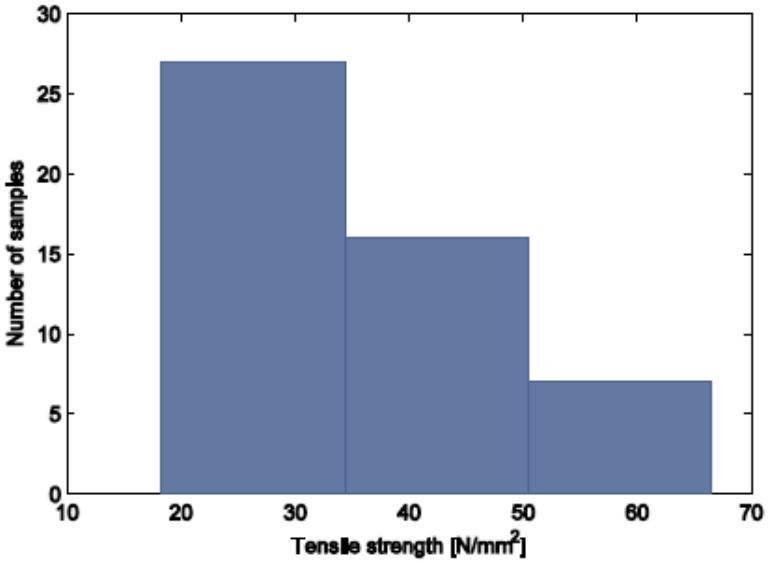
Discrete intevals

Graphical representation of data sets

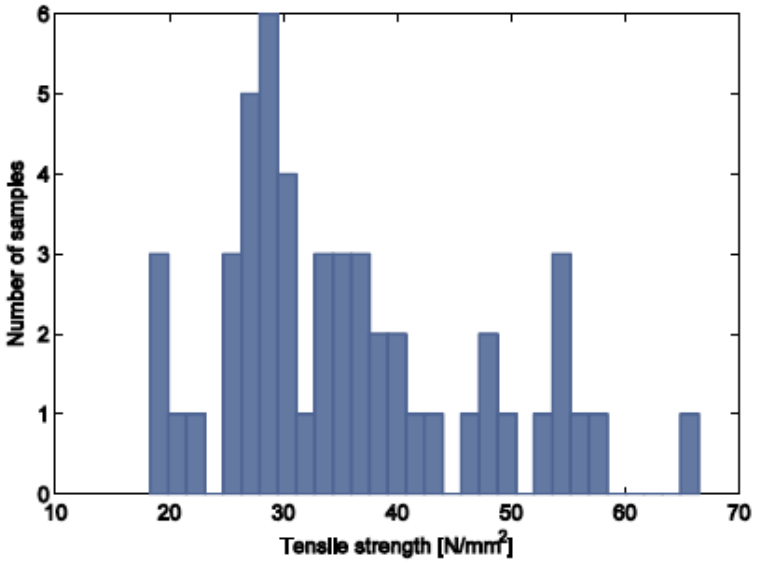
Histogram

- Use intervals of equal width for better interpretation
- Careful selection of number of intervals – use available empirical formulas

Too few intervals



Too many intervals

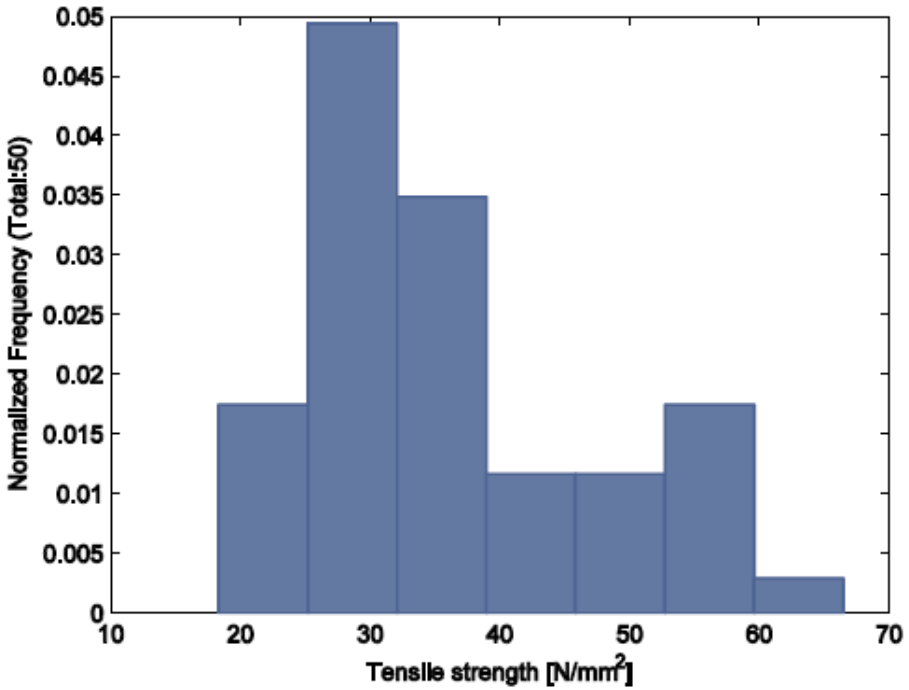


Graphical representation of data sets

Normalized frequency diagram – Obtained by normalization of histogram

$$h_i = \frac{n_i}{n \cdot r_i}$$

- n_i are the number of samples in interval i
- r_i is the width of interval I
- h_i is the heigth of each bar



Normalized frequency diagram of strength data

Graphical representation of data sets

Cumulative frequency diagram – Frequency $Q(x)$ of samples whose values are less than the value x

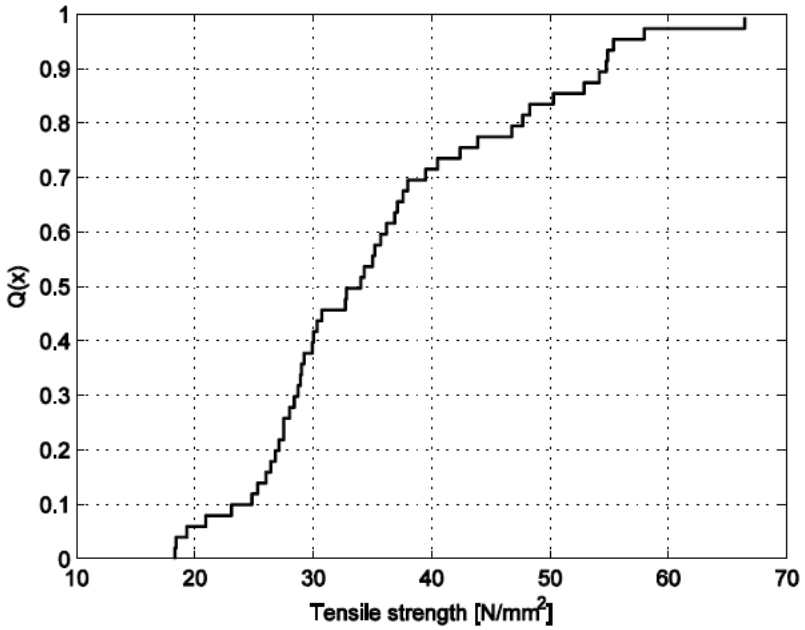
- Sort samples in ascending order
- For x, $Q(x)$ is the number of samples less than or equal to x divided by the total number of samples

```
cdfplot(data)
```

Graphical representation of data sets

Cumulative frequency diagram – Frequency $Q(x)$ of samples whose values are less than the value x

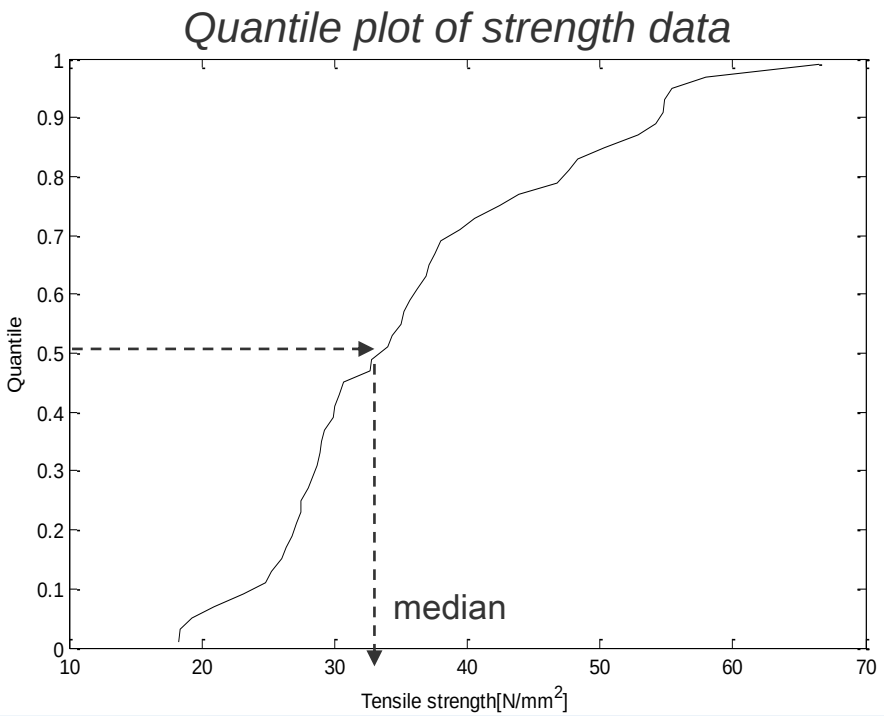
Cumulative frequency diagram of strength data



Graphical representation of data sets

Quantile plot – Plots the values below which a certain fraction of the samples fall

- Sort samples in ascending order $x_1, x_2, \ldots, x_n$
- Plot the pairs x_i and the *i*-th quantile $(i - 0.5)/n$



Graphical representation of data sets

Box plot – Multiple information

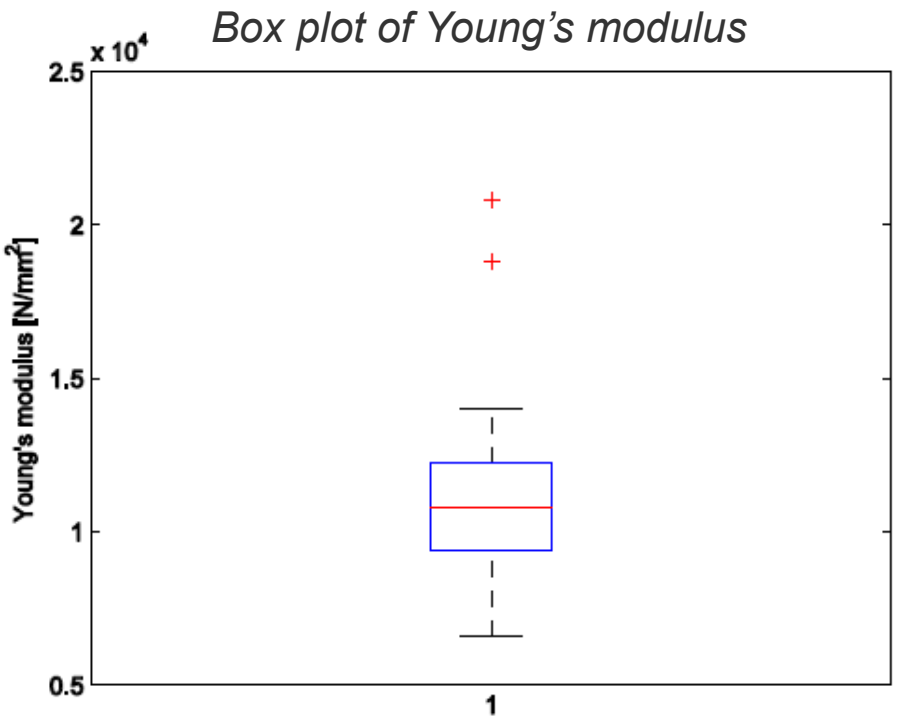
- Quartiles (25%, 50% and 75% percentiles)
- Range (minimum and maximum, excluding outliers)
- Outliers

```
boxplot(data)
```

Graphical representation of data sets

Box plot – Multiple information

- Quartiles (25%, 50% and 75% percentiles)
- Range (minimum and maximum, excluding outliers)
- Outliers



Graphical representation of data sets

Outliers – Possible procedure for determination

- Determine the interquartile range `iqr` (difference between first and third quartile)
- Determine samples greater than `1.5*iqr` above the third quartile
- Determine samples smaller than `1.5*iqr` bellow the first quartile

Pairs of data sets

Scatter diagram – Demonstrates dependence between measured quantities

- Observed quantities in each axis
- Sample represented by a dot

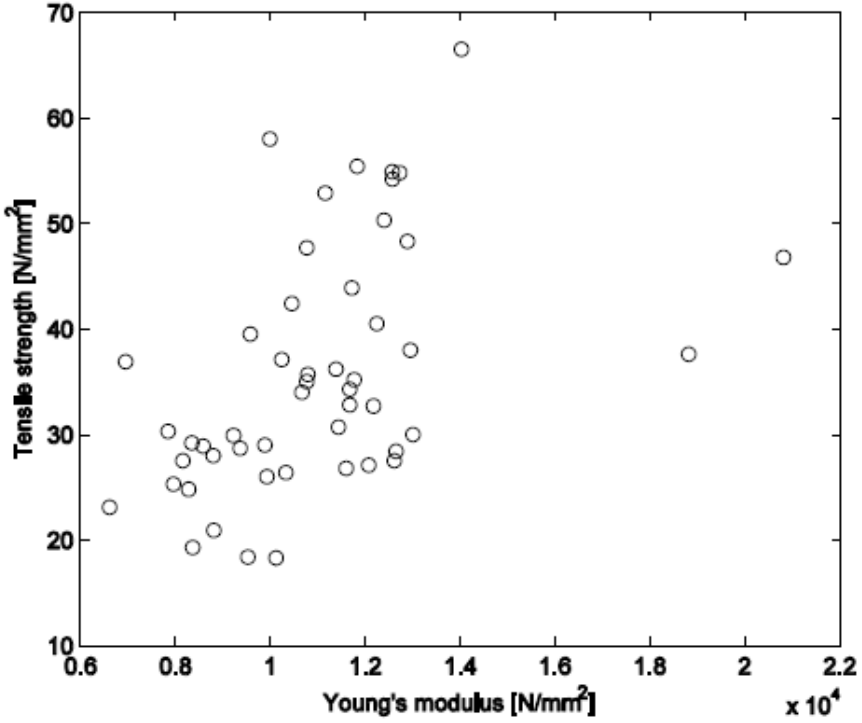
```
scatter(data1,data2)
```

Pairs of data sets

Scatter diagram – Demonstrates dependence between measured quantities

- Observed quantities in each axis
- Sample represented by a dot

Scatter diagram of timber data



Pairs of data sets

Sample covariance – Average of product of deviations from sample mean

$$c_{XY} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

```
cov(data1,data2)
```

Sample correlation coefficient – Normalized covariance

$$r_{XY} = \frac{c_{XY}}{s_X \cdot s_Y}$$

```
corrcoef(data1,data2)
```

Pairs of data sets

Sample correlation coefficient for different pairs of data sets